Harnessing the Human Phenotype Ontology to Predict the Age of Onset of Rare Genetic Diseases

Kyran Wissink[1, 2], Leonardo Chimirri, Ph.D.[2], Kristin Köhler[2], Daniel Danis, Ph.D.[2], and Peter Robinson, MD[2]

[1]Utrecht University, Utrecht, the Netherlands
[2]Berlin Institute of Health at Charité, Berlin, Germany

Internship report

1 Abstract

In this research, we employ machine learning techniques to predict the age of onset of rare genetic diseases using the Human Phenotype Ontology. Providing age of onset information to rare genetic diseases may assist clinicians in differential diagnosis of patients, by narrowing down results.

We first employed a random forest regression model, followed up by a graph convolutional network in an effort to capture temporal traits and nuanced relationships within the dataset. While both models performed well above the baseline with a top-1 accuracy of 85 and 84%, respectively, we failed to identify a significant increase in performance of the neural network over the random forest model.

With this research, we highlight the potential of machine learning in differential diagnostics by capturing the relationship between phenotypic traits and disease progression.

2 Layman's summary

When a patient with a rare disease comes to a doctor's office, it is difficult for the doctor to diagnose this patient. There are thousands of diseases this patient could be suffering from, and many of these diseases present in similar ways. Trying to diagnose this patient is akin to finding a needle in a haystack. This is where the Human Phenotype Ontology (HPO) comes in. This is a large library, containing many rare diseases and their symptoms. To assist with diagnosing such a patient, a doctor can use the HPO database to help narrow down which disease ails the patient, by searching for diseases that fit the description of the patient.

With so many diseases, every bit of information that helps narrow down the possible diseases for the doctor helps. One consistent piece of information that the doctor will have is the age of the patient. However, only around half of the recorded diseases in the HPO have recorded information of the age of patients at the start of the disease. We aimed to provide this data, by making computer models that learn from the existing data and associate an age with each disease.

In this study, we show that these computer models can accurately predict the age of onset of rare diseases based on their symptoms. This can help doctors in diagnosing patients, leading to faster and more effective treatments.

3 Introduction

The Human Phenotype Ontology (HPO) aims to standardise the vocabulary of phenotypic abnormalities observed in hereditary diseases [**robinson-2008**, **gargano-2023**]. The HPO is under continuous development, using medical literature and online databases such as Orphanet, DECIPHER, and OMIM. Each disease is annotated with Human Phenotype (HP) terms, providing a comprehensive database of phenotypic characteristics associated with rare genetic diseases.

Clinicians utilise the HPO as a differential diagnostic tool to assist in diagnosing patients by filtering the ontology for diseases matching the phenotype of the patient [**kohler-2020**]. Given the expanding size of the HPO, which now contains over 10,000 HP terms and annotations for more than 12,000 hereditary diseases, it is imperative to include as much information as possible to narrow down results. One key piece of information that clinicians often have is the age of onset; however, only about half of the diseases in the HPO database include this annotation.

The goal of this research is to append age of onset annotations to more diseases, by employing machine learning algorithms. Accurately predicting the age of onset of rare genetic diseases is crucial, as it can aid clinicians in making differential diagnoses by narrowing down the possibilities based on the patient's age. This can lead to faster and more effective treatments, as early diagnosis is often critical for managing rare diseases [**long-2022**, **rareportal**]. This research can moreover contribute to a better understanding of the relationship between phenotypic traits and disease progression, potentially leading to the development of new treatments and prevention strategies.

We hypothesise that the HPO term annotations of a disease can be leveraged as a predictive factor for the age of onset of a disease since certain phenotypical traits are primarily found within a specific age group. To investigate this, we employ two machine learning models: a **random forest regression model** and a **graph convolutional network**.

The random forest model uses decision trees to predict the age of onset of a disease based on presence or absence of HPO terms. The output of the decision trees is aggregated into the final prediction of the model. In contrast, the GCN learns associations between diseases based on HPO term similarity and

the relative predictive power of these HPO terms, enabling the neural network to make associations that may be missed by the simpler regression model. We hypothesise that the GCN model, with its ability to capture these intricate relationships between the diseases, will outperform the RF model.

We therefore propose the following two hypotheses:

- **H1**: Machine learning approaches may be employed to predict the age of onset of OMIM diseases based on HPO terms.
- **H2**: The GCN model will outperform the RF model.

3.1 HPO

The Human Phenotype Ontology contains standardised terms describing human phenotypical traits. The HPO is a hierarchical graph: the further down the graph is traversed, the more specific the HPO terms become. Any term above an HP term in the hierarchy are called the **ancestor terms**. Likewise, all HPO terms below a term are called the **descendant terms**. There can be multiple degrees of ancestry and descendancy. For illustrative purposes, let us take the term for *polydactyly* (HP:0010442). In our example, the first level ancestor for polydactyly is *Abnormal digit morphology* (HP:0011297). Polydactyly has multiple descendant terms, including *hand polydactyly* and *foot polydactyly*. Going down six more descendants, we eventually end up at *Partial duplication of the proximal phalanx of the 4th finger*. This term does not have any descendants. Ancestry for all HPO terms in the graph eventually leads to the top term: *All* (HP:000001). Between *Partial duplication of the proximal phalanx of the 4th finger* and *All* are fourteen HP terms.

Logically, certain HPO terms are more predictive of an age of onset than others. For example, polydactyly (HP:0010442) can only be congenital, whereas a bone fracture (HP:0020110) can occur at any age. This logic serves as the base of the research.

3.2 Random forest

At the core of random forest models are the decision trees [**breiman-2001**]. Multiple decision trees are trained using **bootstrap aggregating** and the final output of the model is the *leaf* or *node* selected by the most trees. In our case we employ a random forest regression model, where the output is an aggregate of the predictions of all trees, rather than the most predicted class as is the case in classification models.

3.3 Bagging and random subspace method

Bagging or bootstrap aggregating is the general method of training for decision trees [**breiman-1996**]. A subset of the training data is used to train a single tree. Following the training, all the trees receive the same input datapoint, and the class predictions are aggregated as a result.

The random subspace method is an extension to bagging, where, if during training a feature is identified as a strong predictor, this feature will be included in many of the trees in the next training iteration, causing correlation [**ho-1998**]. This is especially important in our case, where certain HPO terms will be highly correlated to a certain age of onset, whereas other HPO terms will be noisy.

3.4 Graph convolutional network

The graph convolutional network (GCN) is a type of graph neural network (GNN): an adaptation of convolutional neural networks (CNN) to graph-like data [**kipf-2016**]. GNNs, as the name suggests, are optimised for processing data represented in graphs. CNNs, on the other hand, are optimised for grid-like data [**zhang-2019**, **sanchez-lengeling2021a**]. We utilise a neural network here, hoping to capture relationships that the random forest model is unable to capture.

3.4.1 Forward function

The data from the batch is then passed to the forward function. The forward function defines how the data is passed through the GCN layers, and returns the onset category predictions for the batch.

The data is first passed to the GCN input layer. After that, the ReLU activation function is used to introduce non-linearity in the model [**hara-2015**]. To prevent overfitting during training, a percentage of neurons are dropped out [**baldi-2013**].

The resulting activations are subsequently fed to the second GCN layer, after which the output is returned.

3.4.2 Back-propagation

Back-propagation is the core of neural network training. It utilises the the chain rule from calculus, to propagate the loss L backward through the network, layer by layer. The following gradients computed by the back-propagation process, instruct the optimiser on how to adjust the network's parameters to minimise the loss [**rojas-1996**]. The model's parameters are then updated with the calculated gradients. Following this process, the calculated gradients are removed to avoid accumulating gradients from previous, and make each training routine independent from the last.

3.5 Application in this research

The random forest model predicts the age of onset of each disease separately, using the aggregated results of the decision trees, while the GCN predicts the age of onset of diseases by learning associations between diseases based on shared HP terms. A basic overview of the machine learning models used in this research is shown in Figure 1.

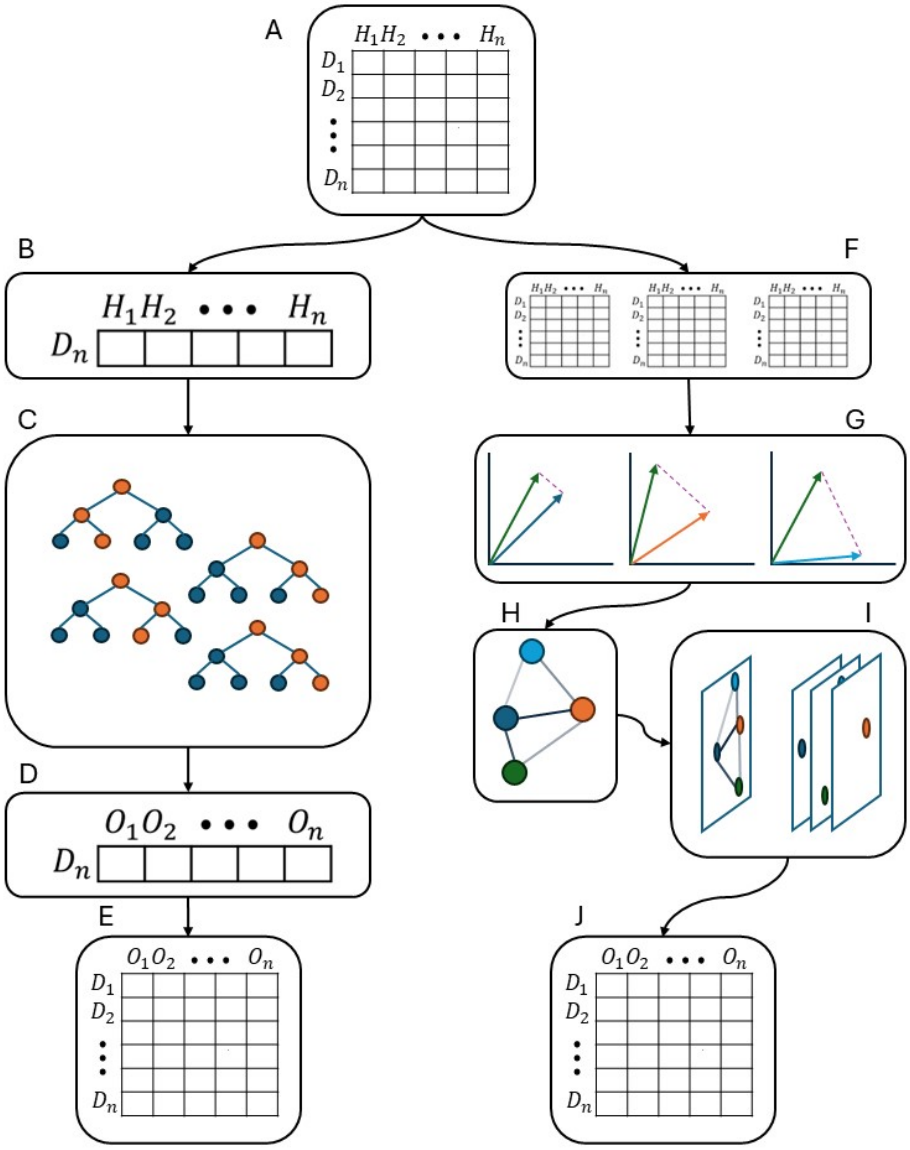


Figure 1: Basic overview of the machine learning models. The random forest model is described on the left side (A, B, C, D, E), while the GCN model is described on the right side (A, F, G, H, I, J). **A**: Data input as a table, with diseases as rows and HPO terms as columns. **B**: The random forest model processes each disease seperately. **C**: Multiple decision trees predict the onset of the disease. **D**: The results of all the trees are aggregated and averaged. **E**: This is repeated for every disease until the data is exhausted, providing onset predictions for every disease. **F**: The data is split into batches. **G**: The similarity between diseases is calculated. **H**: The graph is built using the similarity as connection strengths. **I**: Graph convolution is performed. **J**: This is repeated for every batch until the data is exhausted, providing onset predictions for every disease.

4 Methods

4.1 Data obtaining

The initial dataset was obtained by filtering the HPO database for all OMIM diseases [**hpo-toolkit**, **gargano-2023**]. There are a total of **8182** OMIM diseases in the HPO database, of which **4184** had age of onset annotations.

4.1.1 Onset categories

There are a total of 22 age of onset categories in the HPO database, of which four do not have any disease associations (puerpural, embryonal, perimenopausal, and postmenopausal onset). Of the remaining 19 onset categories, 18 have an allocated temporal time-frame. The only onset category that does not span a range of time is the congenital onset category. This category is specific for phenotypic abnormalities diagnosed at birth. Five of the onset categories are ancestors of other onset categories, and are henceforth referred to as **umbrella terms**. The temporal ranges of onset categories and the hierarchy of the umbrella terms are shown in Figure 2.

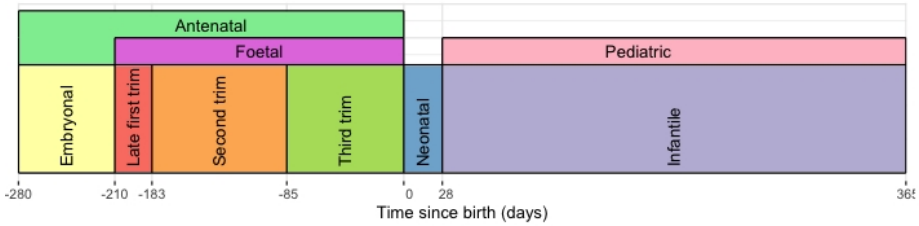
Table 1 shows the number of occurrences of onset categories in OMIM disease annotations. The table furthermore shows the occurrence of umbrella categories.

Onset category	**HPO term**	**OMIM associations**
Late first trimester	HP:0034199	7
Second trimester	HP:0034198	42
Third trimester	HP:0034197	42
Congenital	HP:0003577	1203
Neonatal	HP:0003623	476
Infantile	HP:0003593	1419
Childhood	HP:0011463	960
Juvenile	HP:0003621	730
Early young adult	HP:0025708	62
Intermediate young adult	HP:0025709	22
Late young adult	HP:0025710	46
Middle age	HP:0003596	256
Late	HP:0003584	99
Umbrella categories		
Antenatal onset	HP:0030674	54
Foetal onset	HP:0011461	112
Pediatric onset	HP:0410280	3
Young adult onset	HP:0011462	481
Adult onset	HP:0003581	261

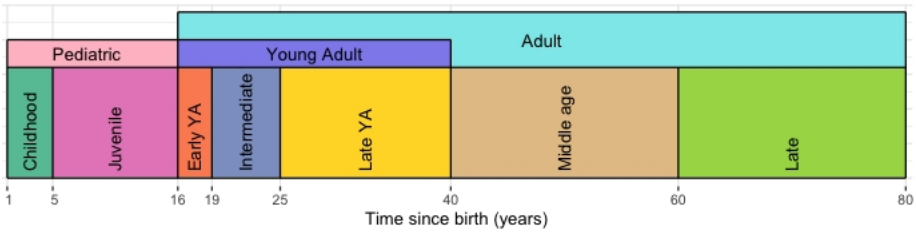
Table 1: Occurrence of onset categories in OMIM disease annotations.

As visible, some categories are relatively underrepresented. This is especially true for the specific antenatal categories. These were hence grouped into their umbrella term: **antenatal onset**.

Table 1 furthermore shows that umbrella categories are responsible for 911 direct annotations, rather than more specific descendant terms. Adult and young adult onset are more often used than their descendant terms, and young adult onset is a descendant term of adult onset. Unlike the antenatal onset treatment described above, we instead opted to distribute onset frequency over the descendants of the umbrella terms to the onset class in the middle of its distribution to minimise loss of information.



(a) Onset categories from gestation to the first year of life.



(b) Onset categories from first year of life onward.

Figure 2: Temporal visualisation of HPO onset categories.

4.2 Disease data preprocessing

The HPO data went through multiple preprocessing steps to make it simpler to use in the models. The OMIM diseases with onset annotation were used for training and validating the model: the **evaluation set**. Each entry in this dataset represents one OMIM disease, and contains the following data:

- `disease_id`: The OMIM identifier for the disease
- `hpo_terms`: A dictionary containing the HPO terms associated with the disease, as well as the associated frequency. This also includes ancestor terms.

- `onset`: An ordered list of onset categories, with frequencies normalised to be $\sum_{c=0}^{O} c = 1$ for every onset category $c \in O$.

4.3 Dataset splitting

A 70/15/15 split was performed on the evaluation set to split the data into a training (n = 2928), test (n = 628), and validation (n= 628) set, respectively. The splitting was performed using the `StratifiedShuffleSplit` from the `sklearn` package, to obtain a representative and relatively equal number of onset categories per dataset and thereby reduce training bias [**pedregosa2011scikit**].

4.4 Random forest model definition

The random forest model was defined using the `RandomForestRegressor` from the Scikit-learn python package [**pedregosa2011scikit**]. The regression algorithm was chosen over the classification algorithm, since we require the model confidence values for evaluation. A grid search for optimal parameters was performed, and the parameters used can be found in Supplementary Table 4.

4.4.1 Model output

Since the number of diseases per tree was set to n = 100, we obtained $B = 30$ decision trees. Each decision tree b makes a classification per disease, and the output of these decision trees is subsequently aggregated and averaged:

$$\hat{y}_d(B) = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_d(b) \tag{1}$$

Whereby $\hat{y}_d(B)$ is the prediction of the random forest for one disease d, and has a value $\hat{y}_o$ for every onset category $o \in O$:

$$\hat{y}_d = \left[\hat{y}_0, \hat{y}_1, \ldots \hat{y}_c\right] \tag{2}$$

This is done for every disease d in the test dataset, thereby obtaining:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}_{00} & \hat{y}_{01} & \cdots & \hat{y}_{0c} \\ \hat{y}_{10} & \hat{y}_{11} & \cdots & \hat{y}_{1c} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{d0} & \hat{y}_{d1} & \cdots & \hat{y}_{dc} \end{bmatrix} \tag{3}$$

4.5 Graph Convolutional Network Model definition

4.5.1 Feature engineering: HPO term weight

In this section, we define how to calculate a feature for each Human Phenotype Ontology (HPO) term, which will be used in our GCN. This feature reflects the predictiveness factor of each HPO term based on its relationship with disease onset categories. Let us define the variables first:

- D: set of all diseases in the evaluation set.
  - d : a single disease in D.
- H: set of all HPO terms in the evaluation set.
  - h: a single HPO term in H.
- O: set of all onset categories for a disease.
  - o: a single onset category of O.

First, we need to quantify how often each onset category occurs in the dataset. For each disease $d \in D$, let μ_d be a function that maps each onset category $o \in O$ to its frequency of occurrence in disease d:

$$\mu_d(o) = \mu(o \in O_d) \tag{4}$$

Then, to calculate the frequency of an onset category $o \in D$, we can use:

$$\forall o \in O: \quad \mu_D(o) = \frac{\sum_{d \in D} \mu_d(o)}{|D|} \tag{5}$$

We do a similar calculation to calculate the frequency of an HPO term $h \in D$. Again, let μ_d be a function that maps each HPO term $h \in H$ to its frequency of occurrence in disease d:

$$\mu_d(h) = \mu(h \in H_d) \tag{6}$$

And:

$$\forall h \in H: \quad \mu_D(h) = \frac{\sum_{d \in D} \mu_d(h)}{|D|} \tag{7}$$

Now, we can continue to calculating the **co-occurrence frequency** between an HPO term h and an onset category o: $\mu(h, o)$, by multiplying an HPO term frequency in a disease $\mu_d(h)$ by the onset category frequency in a disease $\mu_d(o)$ and dividing over the total number of diseases $|D|$

$$\mu(h, o) = \frac{\sum_{d \in D} \mu_d(h) \cdot \mu_d(o)}{|D|} \tag{8}$$

We can subsequently use this information to calculate the **pointwise mutual information** (PMI) between h and o:

$$\text{pmi}(h, o) = \log_2 \frac{\mu(h, o)}{\mu_D(h) \cdot \mu_D(o)} \tag{9}$$

To normalise these values, we calculate the **normalised PMI** (nPMI):

$$\text{nPMI}(h, o) = \frac{\text{pmi}(h, o)}{-\log_2 \mu(h, o)} \tag{10}$$

Now, for each HPO term h, we have a set of nPMI values corresponding to each onset category:

$$\forall h \in H, \exists \Phi = \{\phi(o) \mid o \in O\} \tag{11}$$

These nPMI values indicate how predictive an HPO term is for the age of onset. A term is a strong predictor if its nPMI values are unevenly distributed.

Finally, we calculate the HPO term weight by considering:

- Φ: Set of all nPMI values for h.
- α: The maximum nPMI value in Φ.
- δ: The distance from α to each nPMI value.
- ρ: A penalty factor.

$$\forall \phi(o) \in \Phi : \begin{cases} \text{penalty} \leftarrow \text{penalty} + (\phi(o) \cdot \delta)^{\rho}, & \text{if } \phi(o) > 0, \\ \text{reward} \leftarrow \text{reward} + |\phi(o) \cdot \delta|^{\rho}, & \text{if } \phi(o) \leq 0. \end{cases} \tag{12}$$

Which we can then use to finally calculate the HPO term weight:

$$\omega_h = \frac{\alpha \cdot \text{reward}}{\text{penalty}} \tag{13}$$

4.5.2 Model architecture

Following the data preprocessing and feature engineering steps, the GCN model is initialised using Pytorch [**NEURIPS2019˙9015**] using the parameters shown in Supplementary Table 5, following manual parameter optimisation.

4.5.3 HPO term vector initialisation

To convert the HPO data to a format the GCN can operate with, we create a Pytorch tensor of length $|H|$, for each disease: x_d. Then, the tensor is populated with data by multiplying the HPO term frequency μ_h by the HPO term weight ω_h:

$$x_d = [x_1, x_2, \ldots, x_h] \tag{14}$$

with:

$$\forall h \in H : \quad x_h = \begin{cases} \mu_h \cdot \omega_h, & \text{if } h \in d, \\ 0, & \text{if not } h \in d \end{cases} \tag{15}$$

4.5.4 Collating

The training dataset is split into batches containing N diseases each. Each disease n at this point has a feature vector $x_d(h)$: the sparse HPO term vector with embedded frequencies.

The similarity metric should therefore efficiently handle high-dimensional sparse data, with a focus on orientation of the vector within this space, rather than magnitude. For that reason, we settled on cosine similarity:

$$S_c(A,B) := cos(\theta) = \frac{\vec{A_h} \cdot \vec{B_h}}{\|\vec{A_h}\|\|\vec{B_h}\|} = \frac{\sum_{h=0}^{H} A_h B_h}{\sqrt{\sum_{h=0}^{H} A_h^2}\sqrt{\sum_{i=0}^{H} B_h^2}} \tag{16}$$

Where $S_c(A,B)$ is the cosine similarity between disease A and disease B. This is repeated pairwise for every disease in the batch. If the cosine similarity is then over the similarity threshold τ, an edge is created between the disease nodes in the batch graph, where the edge weight will be $S_c(A,B)$. The end result is a graph per batch, made up of HPO term vectors, edge indices, and edge weights.

In each training cycle, the batches are randomly sampled again. Since there is a class imbalance in the onset categories, weighted random sampling is deployed using resampling. This ensures that each onset category is equally represented in the training loop, to reduce bias introduced by over-representation of some onset categories.

4.5.5 Model output

The model output is a logit for each onset category for each disease in the batch. Therefore, the model output $\hat{\mathbf{Y}}$ is a torch tensor of size [N, C], where N is the number of diseases in the batch and C the number of onset categories:

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}_{00} & \hat{y}_{01} & \cdots & \hat{y}_{0c} \\ \hat{y}_{10} & \hat{y}_{11} & \cdots & \hat{y}_{1c} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{y}_{n0} & \hat{y}_{n1} & \cdots & \hat{y}_{nc} \end{bmatrix} \tag{17}$$

Logits can be converted to probabilities and the total probabilities over all categories C in one disease n should sum to one. For that reason, the method of converting a logit to a probability is with the following softmax function:

$$\sigma(\hat{y})_{n,c} = \frac{e^{\hat{y}_{n,c}}}{\sum_{i=0}^{C} e^{\hat{y}_{n,i}}} \tag{18}$$

Where $\hat{y}_{n,c}$ denotes one output logit at position [n, c] in $\hat{\mathbf{Y}}$. Each probability is normalised to the sum of the probabilities, where:

$$\forall \hat{y}_n \in \hat{\mathbf{Y}}, \quad \sum_{c=0}^{C} \sigma(\hat{y})_{n,c} = 1 \quad \wedge \quad 0 \leq \sigma(\hat{y})_{n,c} \leq 1 \tag{19}$$

4.5.6 Loss calculation

After obtaining the model output predictions $\hat{\mathbf{Y}}$, these are compared to the ground truth; the true onset category probability distribution $\mathbf{Y}$:

$$\mathbf{Y} = \begin{bmatrix} y_{00} & y_{01} & \cdots & y_{0c} \\ y_{10} & y_{11} & \cdots & y_{1c} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n0} & y_{n1} & \cdots & y_{nc} \end{bmatrix} \tag{20}$$

Where any $y_{n,c}$ is an actual probability of onset category c in a disease n from the HPO dataset. Therefore, any $y_{n,c}$ follows much the same rules as any $\hat{y}_{n,c}$:

$$\forall y_n \in \mathbf{Y}, \quad \sum_{c=0}^{C} y_{n,c} = 1 \quad \wedge \quad 0 \leq y_{n,c} \leq 1 \tag{21}$$

Choosing the correct loss function for machine learning prediction optimisation is vital. In this case, we have a classification task based on probability distributions, where the ordinality is important. No one previously defined loss function captures this relationship perfectly. Therefore, we opted to combine two existing ones: **cross-entropy loss** (L_c)and **earth-mover distance**(L_e). The cross-entropy measures the overall similarity between the probability distributions, whereas the earth-mover distance, rather, is a measure of distance. This way, we capture the similarity, the distance, and outliers. Both loss functions are multiplied by a weighting factor, supplied by the user.

The cross entropy loss is defined as:

$$L_c = \frac{\sum_{n=0}^{N} l_n}{N}, \quad l_n = -\sum_{c=0}^{C} \log\left[\sigma(\hat{y})_{n,c}\right] \cdot y_{n,c} \tag{22}$$

And the earth-mover's distance is defined as:

$$L_e = \frac{\sum_{n=0}^{N} l_n}{N}, \quad l_n = \sum_{c=0}^{C} \left|\sigma(\hat{y})_{n,c} - y_{n,c}\right| \tag{23}$$

Note that for both of these equations, the softmax equation eq. (18) was substituted in for legibility.

As is shown here in both eq. (22) and eq. (23), l_n is the loss for one disease n. the sum of all $\sum_{n=0}^{N} l_n$ is divided over the batch size N, thereby effectively taking the mean loss over the batch.

Using ω_c and ω_e as user-defined weighting factors for the cross-entropy and earth-mover's distance loss, respectively, the final weight calculation then becomes:

$$L = \omega_c \cdot L_c + \omega_e \cdot L_e \tag{24}$$

Combining these equations and simplifying then gives us the complete loss function for one batch:

$$L = \frac{1}{N}\left[-\omega_c \sum_{n=0}^{N}\sum_{c=0}^{C} \log \sigma(\hat{y})_{n,c} \cdot y_{n,c} + \omega_e \sum_{n=0}^{N}\sum_{c=0}^{C} |\sigma(\hat{y})_{n,c} - y_{n,c}|\right] \tag{25}$$

4.5.7 Overfitting prevention

Early iterations of the model showed a high proclivity for overfitting. To prevent this, we employed the following machine learning techniques: **early stopping** and a **learning rate scheduler**, as well as a **high drop-out rate**.

In case the loss value during the training phase increases, the learning rate of the model will halve. In the occurrence of a stagnation of the loss value during the training phase for at least three iterations in a row, the model will move to the validation phase and skip further training iterations.

4.6 Key model differences

Both models use a form of quantifying the predictive power of HPO terms: the RF model obtains it as *feature importance* using the random subspace method while the GCN model obtains this as a feature with the method described above.

In the RF model, ordinality is not taken into account. Trees are only trained on whether or not they are correct in their prediction. In the GCN, on the other hand, the loss function is specifically weighted towards ordinality.

The parameters used for both models are shown in Supplementary Table 4 and Table 5.

4.7 Evaluation

Due to the indeterministic nature of the models, we chose to evaluate the models on five different seeds: **1, 42, 1337, 9001** and **131415**. Every evaluation metric described in the results will be the average of the outcome of running the models on these seeds.

To quantify the models, we compare the models to a baseline mock dataset, which contains the average frequency over the **validation** set for every disease. For every single disease entry, this mock dataset has the average onset frequency for each onset category.

5 Results

5.1 Test dataset analysis

The frequency of occurrence of each onset category in the test and validation dataset, as well as in the prediction output of both models is visualised in Figure 3. The model was run five times using different seeds as described in the methods. The mean of these predictions over the onset categories was taken, and the standard deviation is visualised as an error bar over the mean prediction values.

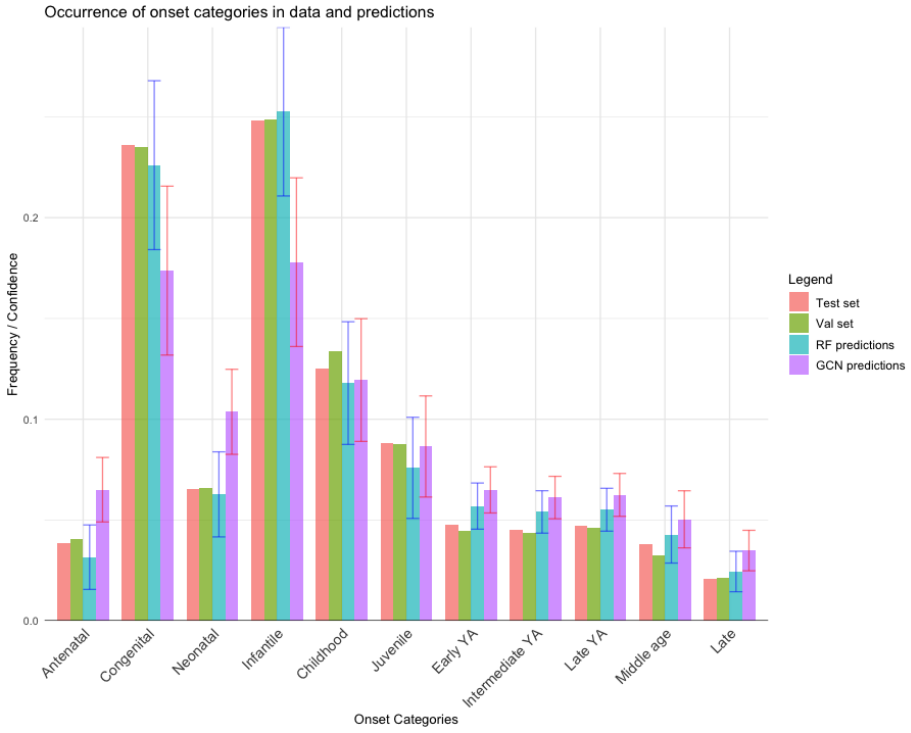


Figure 3: An overview of the frequency and prediction confidence means for all the onset categories. For the predictions, the mean of the results of the runs on five seed were used, and the standard deviation is visualised as an error bar.

5.1.1 Top-1 Accuracy

The top-1 accuracy is a straightforward metric that takes the highest prediction confidence from the model output, and compares that to the truth. In the ground truth, however, multiple answers may be correct. In this section, if the ground truth age of onset frequency for the predicted category is non-zero, it is classified as correct.

The occurrence of ordinal distances to the nearest truth label of the predictions are shown in Figure 4. In the baseline, we get 220 correct predictions, as it always predicts the most common category (Antenatal onset), with an average minimal ordinal distance to a true label of 1.35. The random forest model achieved 394 correct predictions with an average minimal ordinal distance of 0.70. The GCN model achieved 399 correct predictions, with an average minimal ordinal distance of 0.68.

To evaluate the performance difference, Welch two sample t-tests were performed between each pair of the models. The predictions of the RF and GCN model were both significantly better than the baseline (p ¡¡ 0.05). While the GCN performed marginally better in terms of correct predictions and ordinal distance, this effect was not statistically significant (p = 0.69). The models shared the same top-1 prediction for 449 out of 628 diseases (71.5%).

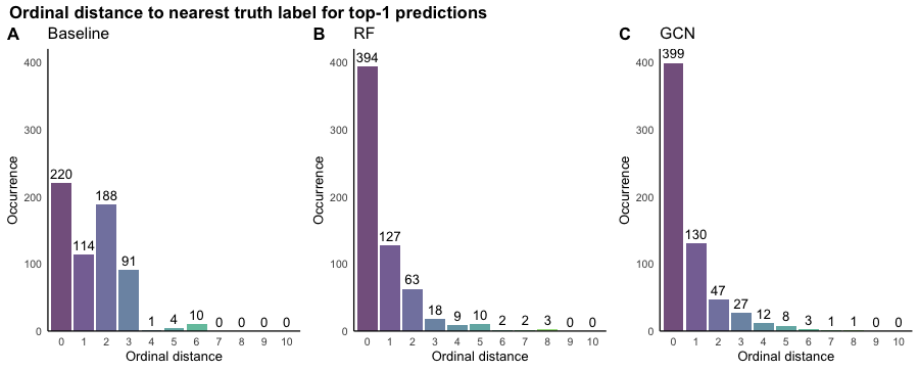


Figure 4: Minimal ordinal distance of top-1 predictions over the test dataset.

5.1.2 ROC analysis

An ROC analysis was performed per model on each of the onset categories in the test dataset. The results of this are shown in Figure 5. Interestingly, both models struggled the most with the childhood onset category. DeLong's tests were performed on the area under the curve (AUC) of all the onset categories between the models (Supplementary Table 2). All DeLong's tests between the baseline and both ML models were significant (p <0.05). None of these were statistically significant between the RF and GCN models (p >0.05). The highest AUC the GCN was able to attain was in the Middle Age category (AUC = 0.9012), while the RF's highest AUC was in the Late category (AUC = 0.8835).

5.2 Precision, Recall, F1

Using the ROC analysis, we selected thresholds to use to consider a prediction valid or not, balancing sensitivity and specificity with. The trade-off between sensitivity and specificity is dependent on the prevalence of the onset in the

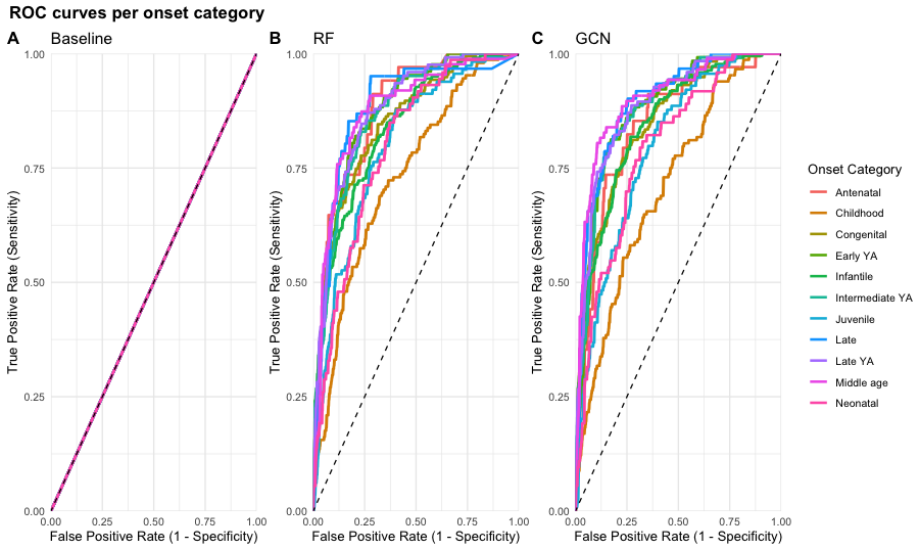


Figure 5: Visualisation of the ROC curves for all of the onset categories.

population. In order to avoid biases, we chose to use the onset prevalence of the validation set, rather than the test or training set.

When applying these threshold values to the predictions, we can compare these to the ground truth and as such obtain evaluation metrics like precision, recall, and F-1 score. However, these metrics do not take into account the ordinality of the data nor the number of onset categories and every single onset of every disease is taken as a datapoint. Table 2 shows the results of these evaluation metrics.

Measure	**RF**	**GCN**	**Derivations**
Sensitivity	0.5641	0.6208	$\frac{TP}{TP+FN}$
Specificity	0.9100	0.8956	$\frac{TN}{FP+TN}$
Precision	0.5893	0.5765	$\frac{TP}{TP+FP}$
Negative Predictive Value	0.9012	0.9116	$\frac{TN}{TN+FN}$
False Positive Rate	0.0900	0.1044	$\frac{FP}{FP+TN}$
False Discovery Rate	0.4107	0.4235	$\frac{FP}{FP+TP}$
False Negative Rate	0.4359	0.3792	$\frac{FN}{FN+TP}$
Accuracy	0.8455	0.8444	$\frac{TP+TN}{P+N}$
F1-Score	0.5764	0.5978	$\frac{2 \cdot TP}{2 \cdot TP+FP+FN}$

Table 2: Evaluation metrics of the test set.

6 Conclusion

Both the random forest and the graph convolutional network models outperformed the baseline. This confirms our first hypothesis that machine learning approaches can be employed in order to predict the age of onset of an OMIM disease based on HPO terms associated with that disease. Our second hypothesis, however, was not confirmed. The GCN model did not outperform the RF model in any manner tested in this research.

7 Discussion

In this research, we showed that machine learning models are capable of predicting the age of onset of OMIM diseases based on HPO terms. The implementation of the work presented here may help clinicians with finding the correct diagnosis for patients with rare genetic diseases faster, and may therefore lead to better disease prognosis [**long-2022**, **rareportal**].

Despite the success in predicting the age of onset with both models, a measurable difference in performance between the models was not observed. Though, we hypothesised that the GCN would perform better than the RF due to the ability of neural networks of finding connections within the dataset that the simpler RF model would not. While this may indicate an absence of these nuanced connections, it may well be due to the relatively low amount of training data. Neural networks have been shown to perform poorly on imbalanced and small datasets [**mazurowski-2007**]. While we did employ resampling to address class imbalance, we did not tackle the dataset size directly. Using diseases from the other disease databases in the HPO was not a valid option, as these have a large overlap in data with the OMIM database. One avenue to explore in the context of buffing the training dataset would be to create simulated data from the training data. This technique is often used for neural networks and moreover prevents overfitting for convolutional networks [**odegaard-2016**, **shorten-2019**].

The apparent converge of both models despite different methodology may also imply other limitations of the dataset. Some diseases may not be well-suited for age of onset prediction. For example, if the disease lacks predictive HPO terms. Due to the structure of age of onset HPO terms outlined in Figure 2 and the subsequent decision to treat umbrella categories as described in the methods, nuance in the data is ostensibly lost. An option for re-assessment would be to work with continuous age of onset, rather than categories.

Many rare genetic diseases are carried by but a few patients. Phenotypic associations with diseases in the HPO may therefore be less reliable, as it may be difficult to differentiate that which is caused by the rare genetic disease and that which is unrelated.

7.1 Application and future direction

The ROC analysis was performed and optimal thresholds per model per onset category were chosen solely for evaluation purposes. This is not the manner in which the output of these models should be utilised in practice. Instead, the model confidence could influence the ordering of the differential diagnosis results given to the clinician. The clinician may then focus on diseases predicted by the models to be more common at that patient's age.

The random forest and convolutional network were used completely separately in this study. However, using a random forest model as a building block to create a neural network has been shown to be a valid approach [**wang-2017**]. This may prove to be an interesting avenue for future exploration.

8 Supplementary

8.1 Software and packages

Module	Version
hpo-toolkit	0.5.0
Pandas	2.2.2
torch	2.3.1
torch_geometric	2.5.3
scikit-learn	1.5.0
numpy	1.26.4
scipy	1.13.0

Table 3: Modules and versions used in this research.

8.2 Parameters

Parameter	Value
Number of trees	100
Criterion	Squared error
HPO threshold	5
Minimum sample split	2
Minimum sample leaf	1
Bootstrap	True

Table 4: RF parameters

Parameter	Value
Optimiser	AdamW
Batch size	32
Hidden channels	32
HPO threshold	5
Penalty factor	2
Dropout rate	0.70
Similarity threshold	0.05
Learning rate	0.001
Weight decay	0.001

Table 5: GCN Default parameters

8.3 ROC, AUC, and DeLong's tests

Category	**AUC(RF)**	**AUC(GCN)**	**DeLong's p-value**
Antenatal	0.878504	0.8468013	0.1089
Congenital	0.8485777	0.8510501	0.8371
Neonatal	0.7980007	0.7921017	0.742
Infantile	0.8371101	0.8476493	0.324
Childhood	0.7311655	0.7121622	0.3737
Juvenile	0.7967609	0.7930405	0.8314
Early YA	0.8763594	0.8781719	0.8329
Intermediate YA	0.8688562	0.8802061	0.2269
Late YA	0.8799283	0.8872728	0.4061
Middle Age	0.8775044	0.9011834	0.05326
Late	0.8834822	0.8963773	0.3473

Table 6: AUC and DeLong's p-value results.